

基于智能客户端技术的售票系统改良方案

The implement of the improved ticketing systems based on Smart - Client technology

刘建伟 林延君 王家宝 (大连理工大学精密与特种加工教育部重点实验室 116023)

摘要:现有的票点售票系统大多基于 B/S 模式,存在许多缺陷,系统功能有限,性能低下。智能客户端技术是一种新型的客户端技术,它结合了胖客户端应用程序的优点与瘦客户端应用程序的部署和可管理性优点。本文分析比较了智能客户端技术的特点,提出了基于智能客户端技术的票点售票系统的改良方案,并通过调用 Web Services 生成了系统框架。

关键词:票点系统 胖客户端 瘦客户端 智能客户端 Web Services

1 引言

随着 internet 技术的发展,基于 B/S 模式的系统以其良好的事务性、易管理性、易部署性等优点在各个行业的信息管理系统得到了广泛的应用,例如作者所讨论的售票管理系统。票点销售在整个票务事务处理过程中占有重要地位,它是票务的重要销售渠道,它还完成包括打印票、各种异常处理、帐务结算等其它关键业务,因此说票点销售系统的成败与否关系到整个票务系统的发展。随着票点系统的业务处理量日益加大和功能需求的不断加强,传统意义上的 B/S 模式的实现也逐渐暴露出弊端。对于规模比较大的票点系统来说,由于本身处理的业务量很大,使用 B/S 模式将不能实时地进行处理,而较长时间的等待,也将会影响客户购票的积极性,最终将会严重影响业务的开展。居于上述受客户端的局限性,为了更好地解决功能、性能、安全性的问题,本文采用了智能客户端模式对票点售票系统进行了改良。智能客户端技术是一种新型的客户端技术,采用基于该技术的客户端模式将胖客户端应用程序的优点与瘦客户端应用程序的部署和可管理性优点结合起来,能很好的解决上述问题。

2 智能客户端简介

要完全了解智能客户端如何将胖客户端与瘦客户端的优点结合起来,有用的做法是分析胖/瘦客户端应用程序模型背后的历史和基础原理,并且回顾一些与

它们相关联的优点和缺点。

2.1 胖客户端应用程序

在二十世纪九十年代中期,为 Microsoft Windows 操作系统开发的胖客户端应用程序的数量急剧增长。设计这些客户端的目的是利用本地硬件资源以及客户端操作系统平台的功能。

尽管胖客户端通常提供了高质量、响应迅速的用户体验,并且具有良好的开发人员和平台支持,但它们非常难于部署和维护。随着应用程序和客户端平台的复杂性不断增加,以可靠且安全的方式将应用程序部署到客户计算机的难度也将不断增加。如果部署了不兼容的共享组件或软件库,则一个应用程序可以很容易地破坏另一个应用程序,这种现象称为应用程序脆弱性。

2.2 瘦客户端应用程序

Internet 提供了传统胖客户端模型的替代模型,它解决了许多与应用程序部署和维护相关联的问题。基于浏览器的瘦客户端应用程序是在中央 Web 服务器上部署和更新的;因此,它们消除了将应用程序的任何部分显式地部署到客户计算机并加以管理的必要性。

瘦客户端应用程序具有一些缺点。浏览器必须总是具有网络连接。因为应用程序的大部分逻辑和状态位于服务器上,所以瘦客户端会频繁地向服务器发回数据和处理请求。浏览器必须等待响应到达,然后用户才能继续使用该应用程序;因此,该应用程序的响应

速度通常要比胖客户端应用程序慢得多。

2.3 智能客户端应用程序

智能客户端应用程序是瘦客户端应用程序的强大替代产品。它们可以为用户提供内容丰富且响应迅速的用户界面,提供脱机工作的能力,并且提供利用本地硬件和软件资源的方法。此外,可以将它们设计为在各种各样的客户端设备上运行,包括桌面 PC、平板 PC 以及手持移动设备。智能客户端为用户提供了在强大且直观的客户端环境中访问信息和远程服务的能力,并且是一种用来开发灵活的、面向用户的应用程序以及提高用户工作效率和满意度的有效的解决方案。

经过设计,智能客户端应用程序可以将胖客户端应用程序的优点与瘦客户端应用程序的部署和可管理性优点结合起来,所有智能客户端都具有下列部分或全部特征:

- (1) 利用本地资源
- (2) 利用网络资源
- (3) 支持偶尔连接的用户
- (4) 提供智能安装和更新
- (5) 提供客户端设备灵活性

2.4 三种客户端程序的比较

传统胖客户端所具有的一些最大的问题发生在部署或更新应用程序的时候。许多胖客户端应用程序具有大量复杂的安装要求,并且可能通过注册组件以及/或者在公共位置安装 DLL 来共享代码,从而导致应用程序脆弱性和更新困难。可以对智能客户端应用程序进行设计,以使其按照远比传统胖客户端应用程序更智能和灵活的方式来管理它们的部署和更新。它们可以避免上述常见问题,从而有助于减少应用程序的管理成本。

设计良好的智能客户端应用程序最大限度地利用了代码和数据部署在客户端上并且在本地执行和访问这一事实。它为应用程序提供了内容丰富且响应迅速的用户界面,以及强大的客户端处理能力。

智能客户端可以利用客户端硬件资源(如电话或条码读取器)以及其他软件和应用程序。这使它们非常适合于解决瘦客户端应用程序(如销售点终端应用程序)无法很好解决的问题。

构建智能客户端的最大的好处就是可以离线使用,离线式工作方式可以在程序重新在线时,自动接

收数据和应用程序更新,而这是瘦客户端所无法实现的。

表 1 列出了智能客户端和胖/瘦客户端在体系结构上的种种不同。

表 1 瘦/胖客户端和智能客户端的特点

特点	瘦客户端	胖客户端	智能客户端
提供内容丰富的用户界面	可以,但难以开发、测试和调试	可以	可以
可以利用本地计算机上的硬件资源	可以,但只能通过 COM 组件使用	可以	可以
可以与其他本地应用程序交互	否	可以	可以
可以多线程化	否	可以	可以
可以脱机工作	否	可以	可以
在低带宽和高延迟环境中可以很好地执行	否	可以	可以
易于部署	是	困难	取决于应用程序要求
低维护和更改管理成本	是	困难	成本取决于应用程序要求
可以部署到具有不同功能的多种客户端上	是	困难	可以,能在支持 .NET 框架的任何平台上部署

3 系统的设计实现

3.1 解决方案体系结构

票点销售系统解决方案包含三个主要组件:数据库、XML Web 服务和使用 Windows 窗体类生成的智能客户端应用程序(见图 1)。

数据库由 XML Web 服务访问,这些 XML Web 服务仅具有在数据库上运行存储过程的权限。通过限制 XML Web 服务在数据库上所能访问的内容,可以确保只有具有相应权限的查询才能运行在数据库上。

XML Web 服务实现为在公共服务器上运行,任何应用程序都可以通过 Internet 对其进行访问。当然,可以在企业的 Intranet 上运行它们,从而将数据访问限制到内部网络。

智能客户端应用程序通过将用户名和密码传递给身份验证 XML Web 服务来对用户进行身份验证。身份验证成功后,将有以下两种数据请求处理方法:其一,XML Web 服务将向智能客户端应用程序传回一个加密票,该加密票将被存储,并在将来每次请求数据时提交给数据 XML Web 服务。数据 XML Web 服务将验

证该加密票并处理数据请求。其二,使用安全的套接字层(SSL)进行敏感数据的传输。

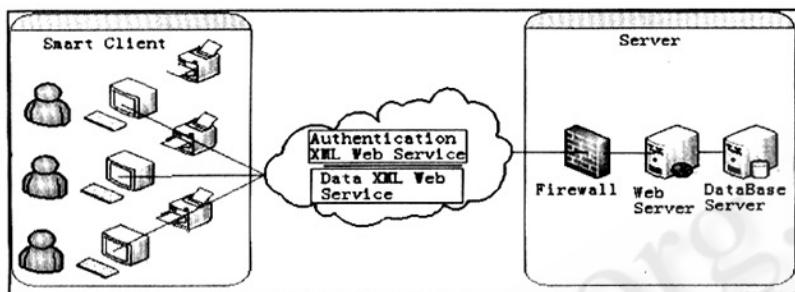


图1 智能客户端票点销售系统体系结构

3.2 几个技术难题的解决

3.2.1 打印方案的比较实现

下面就B/S模式下的两种优秀打印方案与智能客户端打印方案进行比较分析。

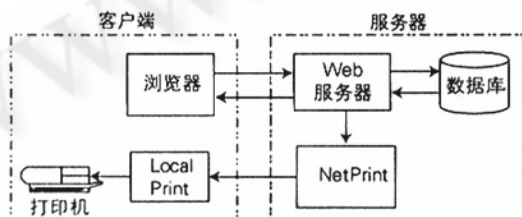


图2 票点票券打印系统模型图

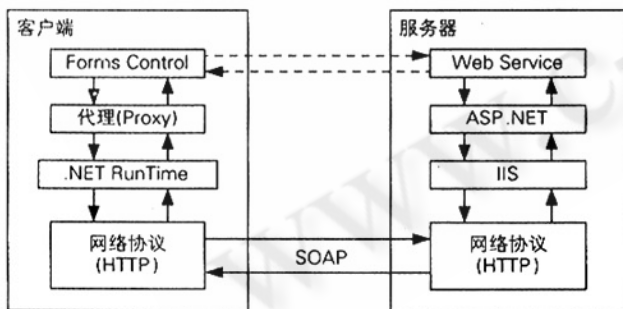


图3 基于Web Services和Forms Controls的打印方案

(1) B/S模式下票券打印方案之一

在Web上实现打印最简单的方案是调用浏览器内置的打印功能,但其功能十分有限,无法精确控制票面文字的位置,也无法灵活地指定字体、字型、字体大小等。而且条形码的打印是票券打印的常用功能,仅凭浏览器自带的打印功能无法做到。

解决上述问题的一种适用方案是采用Socket通讯机制传送票券资料,在客户端用COM组件来实现打印功能。具体描述如下:在客户端运行打印进程(LocalPrint),它既作为Socket的Server端,负责侦听来自网站服务器的信息,又负责精确控制打印。在服务器端嵌入ActiveX控件(NetPrint),其中封装Socket的Client端。当用户向服务器提出打印申请时,服务器端脚本查询数据库得到需要的票券信息,传递给NetPrint。NetPrint与LocalPrint通过基于TCP的Socket通信机制进行数据传输,LocalPrint进行字体等打印细节的控制,实现客户端打印。过程描述如图2所示。

该方案的优点是功能强大,较为安全可靠,并且能打印精度要求较高的条形码。该方案是ASP平台下实现票券打印的优秀方案。但它又存在以下的不足:技术层次相对复杂,编程难度加大。在传输票券数据的时候,需要制定复杂的编码、解码及校验规则。服务器端无法准确监控打印状态。只要Socket通讯建立成功,就认为打印成功,客户端是否真正成功打印无法反馈给服务器。

客户端控制功能有限。客户端被动接收票券资料,无法向服务器端发出控制信息。客户端需要运行程序,负责侦听端口和控制打印。

(2) B/S模式下票券打印方案之二

该方案是基于Web Services的打印方案。它结合微软的.NET框架,在IE中嵌入Windows Forms控件,该控件唤醒一个远程的Web Service,取得封装好的票券资料,然后利用.NET提供的打印类,精确的控制打印。其中的Windows Forms控件是紧凑、轻量级的高安全性的客户端应用程序,它与Java applets的原理相似,性能却优于Java applets。其结构如图3所示。

该方案是基于Web Services的打印方案。它结合微软的.NET框架,在IE中嵌入Windows Forms控件,该控件唤醒一个远程的Web Service,取得封装好的票券资料,然后利用.NET提供的打印类,精确的控制打印。其中的Windows Forms控件是紧凑、轻量级的高安全性的客户端应用程序,它与Java applets的原理相似,性能却优于Java applets。其结构如图3所示。

该方案是基于Web Services的打印方案。它结合微软的.NET框架,在IE中嵌入Windows Forms控件,该控件唤醒一个远程的Web Service,取得封装好的票券资料,然后利用.NET提供的打印类,精确的控制打印。其中的Windows Forms控件是紧凑、轻量级的高安全性的客户端应用程序,它与Java applets的原理相似,性能却优于Java applets。其结构如图3所示。

该方案是基于Web Services的打印方案。它结合微软的.NET框架,在IE中嵌入Windows Forms控件,该控件唤醒一个远程的Web Service,取得封装好的票券资料,然后利用.NET提供的打印类,精确的控制打印。其中的Windows Forms控件是紧凑、轻量级的高安全性的客户端应用程序,它与Java applets的原理相似,性能却优于Java applets。其结构如图3所示。

该方案是基于Web Services的打印方案。它结合微软的.NET框架,在IE中嵌入Windows Forms控件,该控件唤醒一个远程的Web Service,取得封装好的票券资料,然后利用.NET提供的打印类,精确的控制打印。其中的Windows Forms控件是紧凑、轻量级的高安全性的客户端应用程序,它与Java applets的原理相似,性能却优于Java applets。其结构如图3所示。

该方案是基于Web Services的打印方案。它结合微软的.NET框架,在IE中嵌入Windows Forms控件,该控件唤醒一个远程的Web Service,取得封装好的票券资料,然后利用.NET提供的打印类,精确的控制打印。其中的Windows Forms控件是紧凑、轻量级的高安全性的客户端应用程序,它与Java applets的原理相似,性能却优于Java applets。其结构如图3所示。

该方案是基于Web Services的打印方案。它结合微软的.NET框架,在IE中嵌入Windows Forms控件,该控件唤醒一个远程的Web Service,取得封装好的票券资料,然后利用.NET提供的打印类,精确的控制打印。其中的Windows Forms控件是紧凑、轻量级的高安全性的客户端应用程序,它与Java applets的原理相似,性能却优于Java applets。其结构如图3所示。

该方案是基于Web Services的打印方案。它结合微软的.NET框架,在IE中嵌入Windows Forms控件,该控件唤醒一个远程的Web Service,取得封装好的票券资料,然后利用.NET提供的打印类,精确的控制打印。其中的Windows Forms控件是紧凑、轻量级的高安全性的客户端应用程序,它与Java applets的原理相似,性能却优于Java applets。其结构如图3所示。

该方案是基于Web Services的打印方案。它结合微软的.NET框架,在IE中嵌入Windows Forms控件,该控件唤醒一个远程的Web Service,取得封装好的票券资料,然后利用.NET提供的打印类,精确的控制打印。其中的Windows Forms控件是紧凑、轻量级的高安全性的客户端应用程序,它与Java applets的原理相似,性能却优于Java applets。其结构如图3所示。

该方案是基于Web Services的打印方案。它结合微软的.NET框架,在IE中嵌入Windows Forms控件,该控件唤醒一个远程的Web Service,取得封装好的票券资料,然后利用.NET提供的打印类,精确的控制打印。其中的Windows Forms控件是紧凑、轻量级的高安全性的客户端应用程序,它与Java applets的原理相似,性能却优于Java applets。其结构如图3所示。

该方案有以下优点:

Web Services 技术的运用规范了票务数据传输的规则,提高了安全机制,而且便于日后异构平台上客户端打印的扩展。

服务器端能准确监控客户端的打印状态,一旦打印失败,系统会终止打印进程,不会写入错误的记录。

可以在客户端控制打印,实现 Fat-Client 应用。

但是仍存在以下不足:

由于局部调用了 Web Service,增加系统设计的复杂性。

处理过程复杂,超长的票点流水号不能直接进行赋值于 Windows Forms 控件,而只能由工作人员手工输入。

在 IE 中嵌入 Windows Forms 控件,打印速度缓慢。

(3) 智能客户端模式下票券打印方案

此方案同样是基于 Web Services 的打印方案,因此它具有上述方案二的优点。但是,智能客户端本身就是一个窗体程序,直接就可以进行打印,故打印速度快;全局调用 Web Service,没有系统设计的复杂性这一说法;票点流水号等票券资料直接就可以传入打印窗体,处理过程简单。

3.2.2 智能部署客户端程序

易部署性是瘦客户端应用程序取代胖客户端应用程序的一个很主要的特征,如果想用新的模式将之替代,那么必须得继承这个特点。在这里,使用了微软的 Updater Application Block 组件,该应用程序块带有应用程序更新存根的无接触部署支持应用程序的事务性安装。Updater Application Block 可帮助确保应用程序成功地完整安装。要执行事务性安装,您将需要包含相应的代码,以便除了执行自动更新以外,还检查是否已经在本地硬盘上安装了所有代码。

可以设置定期检测,也可以设定启动时检测,这里采取启动检测的方式。以下是它的工作方式:

启动应用程序——请求服务器检查是否有新版本——如果有则提示下载更新(否则不作动作)。

为了防止服务器端和客户端进行交互的时候被窃听,更新到某种有巨大破坏力的攻击文件,故需要通过加密传输机制。这里专门定义了一个类,在请求回应

的时候,由服务器端进行加密,然后到客户端进行解密,解密后比对特殊字符串,如果相符就进行下载更新动作,如果不符,则表示是外界不安全代码,拒绝更新。

通过将无接触部署与应用程序更新存根结合起来,简化了部署和更新,并且使得应用程序运行在完全受信任的环境中。

4 结论与展望

本文应用了智能客户端方案构架了票点销售系统,集胖瘦客户端的优点于一身的智能客户端由于自身的优点,在票务设计领域中将逐渐取代瘦客户端模式,成为新一代的客户端/服务器端交互模式。事实证明,由于票务领域都具有共性,使得这一解决方案都普遍适用于各式的分布式票点系统,所不同的只是业务逻辑的差异而已。

由于在不同的票务领域中,智能客户端的本地化资源、网络化资源的利用程度都不一样,以及它还具有支持偶尔连接的用户、提供客户端设备灵活性等优点。因此,在实际的票点系统设计过程中,结合具体的需求,充分利用智能客户端的各种特性,量身订造出适合自身要求的票务系统是今后票点设计的发展趋势。

参考文献

- 1 Mark Boulter, 智能客户端体系结构与设计指南, 2004.8-20。
- 2 柴晓路、梁宇奇, Web Services 技术、架构和应用. 1, 北京 电子工业出版社, 2003.3-25。
- 3 胡延平、唐清华等, 广域网票券客户端打印的解决方案研究与实现, 计算机工程与应用, 2003, 7: 230-232。
- 4 金敬东, 应用数字证书的网上订票系统[C], 北京理工大学学位论文, 2001, 9。
- 5 朱辉生, 基于 WindowsDNA 网上购物系统的研究与开发[C], 河海大学博士学位论文, 2001, 7: 14-16。
- 6 石伟鹏、杨小虎, 基于 SOAP 协议的 Web Service 安全基础规范(WS Security), 计算机应用研究, 2003, 2: 100-102。